REMARKS/ARGUMENTS

Overview of the Office Action

Claims 1, 2, and 8 were rejected under 35 U.S.C. § 102(e) as being anticipated by Farrell et al. (U.S. Patent No. 6,269,475).

Claims 3-7, 9, 21, and 28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Washburn et al. (U.S. Patent No. 5,157,779).

Claims 10-12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Bier et al. (U.S. Patent No. 5,133,052), Williamson et al. (U.S. Patent No. 6,122,641), and Peddada et al. (U.S. Patent No. 6,031,533).

Claim 13 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Gupta et al. (U.S. Patent No. 6,484,156).

Claim 14 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Gupta et al. (U.S. Patent No. 6,484,156) and O'Donnell et al. (U.S. Patent No. 6,223,203).

Claims 15-20 and 22-27 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Chow et al. (U.S. Patent No. 5,975,334).

Status of the Claims/Amendments

Claims 1, 15, and 22 have been amended. Claims 1-28 are pending.

Explanation of Amendments to the Claims

In regard to Claims 1, 15, and 22, these claims has been amended to clarify that the "graphical object" of said claims represents the actual code of the source code module. These amendments introduce no new matter, and the subject matter of the claims is supported by and described in the Specification of the present Application (see, e.g., Specification, page 2, line 31 to page 3, line 1, as well as page 3, lines 5-6). In this regard, it is the sole intention of the Applicants to narrow these claims (and, indirectly, the claims that depend thereon) only to the minimal extent necessary to avoid the teachings of Farrell et al. (U.S. Patent No. 6,269,475), and nothing in these amendments should be interpreted or construed as disclaiming any of the scope or breadth of the original claims except to the extent such original claims are specifically anticipated by the teachings of the Farrell reference alone.

Claims Rejected Under 35 U.S.C. § 102(e)

Claims 1, 2, and 8 were rejected under 35 U.S.C. § 102(e) as being anticipated by Farrell et al. (U.S. Patent No. 6,269,475). In response, Applicants have amended Claim 1, upon which Claims 2 and 8 depend, to clarify that the "graphical object" of said claim represents the actual code of the source code module.

The invention of Farrell pertains to "[a]n object oriented program editor operative on a text source in a language having syntax properties having a lexical analyzer, a parser, a codeblock generator and a graphical user interface" (Farrell, Abstract, lines 1-4). The system of Farrell is summarized in its abstract, which states:

> The lexical analyzer identifies language tokens in the text source. The parser which is coupled to the lexical analyzer associates syntax properties with the tokens. The codeblock generator which is coupled to the parser, **groups tokens into a tree of codeblocks**. The graphical user interface receives and implements user modifications of the codeblocks **in a manner consistent with the programming language**. A **system for generating source code in a program language with respect to an object model** having a plurality of classes and relations among such classes having a class selection interface, an action selection interface and a source code generator. ... The source code generator is used for **generating source code** in the program language to cause the action to be performed on the entity, and associating such code with the chosen class.

(Farrell, Abstract, lines 4-25; emphasis added.)

In the invention of Farrell—and using terminology consistent with that found in the present Application—modified code is processed by a modeling tool to update the object model and vice versa (Id.). However, as noted in the Specification of the present Application, this approach, like other prior art approaches, "suffers from the fact that the model and the code are two separate elements that are processed to generate one from the other rather than being two views of the same information" (Application Specification, page 2, lines 1-3). Consequently, the Farrell system essentially consists of a text editor, a GUI-based editor, and an intervening application to translate between the two (see, e.g., Farrell, Fig. 2). Consequently, in the

invention of Farrell, the graphical object does not directly represent actual code in the source code module, but merely represents the source code module by aggregation and proxy.

In contrast, the invention of the present Application is directed to "an integrated development environment wherein there is a tight coupling between the design surface providing a visual representation of the various physical and logical entities in the software model and the underlying code structures that support the entities" such that "[e]very object defined within the design surface is capable of being mapped **directly** to an underlying code structure" (Application Specification, page 2, line 27 to page 3, line 1) (emphasis added). The net result is that, in the invention of the present Application, "[t]he model is a graphical representation of **the actual code**" (Application Specification, page 2, line 5) (emphasis added). To clarify this point in the claims, the aforementioned amendments have been made.

In order to anticipate a claimed invention, a prior art reference must teach each and every element present in the claim. In this regard, and in light of the clarification set forth in the amended claims, Applicants respectfully submit that Farrell does not teach the element of a graphical object representing actual code of the source code module such that every object defined within the design surface is capable of being mapped directly to the underlying source code structure. Consequently, Farrell fails to teach or suggest all the claim elements necessary to anticipate the invention of Claims 1 under 35 U.S.C. § 102(e). In addition, claims that depend upon an allowable claim are also allowable, and dependent Claims 2 and 8 depend upon independent Claim 1. Therefore, Applicant(s) respectfully request that the rejections as to Claims 1, 2, and 8 be withdrawn and that these claims be allowed to issue.

Claims Rejected Under 35 U.S.C. § 103(a)

Claims 3-7, 9, 21, and 28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Washburn et al. (U.S. Patent No. 5,157,779). Claims 10-12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Bier et al. (U.S. Patent No. 5,133,052), Williamson et al. (U.S. Patent No. 6,122,641), and Peddada et al. (U.S. Patent No. 6,031,533). Claim 13 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Gupta et al. (U.S. Patent No. 6,484,156). Claim 14 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Gupta et al. (U.S. Patent No. 6,484,156) and O'Donnell et al. (U.S. Patent No. 6,223,203). Claims 15-20 and 22-27 were

rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Chow et al. (U.S. Patent No. 5,975,334).

Regarding Claims 3-7 and 9-28

Claims 3-7 and 9-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of one or more of the following: Washburn et al. (U.S. Patent No. 5,157,779); Bier et al. (U.S. Patent No. 5,133,052); Williamson et al. (U.S. Patent No. 6,122,641); Peddada et al. (U.S. Patent No. 6,031,533); Gupta et al. (U.S. Patent No. 6,484,156); O'Donnell et al. (U.S. Patent No. 6,223,203); and/or Chow et al. (U.S. Patent No. 5,975,334). In response, Applicants have amended independent Claims 1, 15, and 22—upon Claims 3-7 and 9-14; 16-21; and 23-28 depend respectively—to clarify that the "graphical object" of said claims represents the actual code of the source code module.

As previously discussed earlier herein (and repeated here for convenience), the invention of Farrell (the primary cited reference) pertains to "[a]n object oriented program editor operative on a text source in a language having syntax properties having a lexical analyzer, a parser, a codeblock generator and a graphical user interface" (Farrell, Abstract, lines 1-4). The system of Farrell is adequately summarized in its abstract, which states:

> The lexical analyzer identifies language tokens in the text source. The parser which is coupled to the lexical analyzer associates syntax properties with the tokens. The codeblock generator which is coupled to the parser, **groups tokens into a tree of codeblocks**. The graphical user interface receives and implements user modifications of the codeblocks **in a manner consistent with the programming language**. A **system for generating source code in a program language with respect to an object model** having a plurality of classes and relations among such classes having a class selection interface, an action selection interface and a source code generator. ... The source code generator is used for **generating source code** in the program language to cause the action to be performed on the entity, and associating such code with the chosen class.

(Farrell, Abstract, lines 4-25; emphasis added.)

In the invention of Farrell—and using terminology consistent with that found in the present Application—modified code is processed by a modeling tool to update the object model

and vice versa (Id.). However, as noted in the Specification of the present Application, this approach, like other prior art approaches, "suffers from the fact that the model and the code are two separate elements that are processed to generate one from the other rather than being two views of the same information" (Application Specification, page 2, lines 1-3). Consequently, the Farrell system essentially consists of a text editor, a GUI-based editor, and an intervening application to translate between the two (see, e.g., Farrell, Fig. 2). Consequently, in the invention of Farrell, the graphical object does not directly represent actual code in the source code module, but merely represents the source code module by aggregation and proxy.

Moreover, this important shortcoming is not mitigated by combining the teachings of Farrell with the teaching of Washburn, Bier, Williamson, Peddada, Gupta, O'Donnell, Chow, or any combination thereof. Washburn, cited by the Examiner for its description of a data store, is directed to a "user extensible, automated testing system" (Washburn, Abstract, line 1). Bier, cited by the Examiner for describing a graphical object that represents a database object, is directed to methods for "search[ing] digital synthetic graphics data" (Bier, Abstract, lines 1-2). Williamson, cited by the Examiner for binding a variable to a database column, is directed to "a model that maps object classes in an object-oriented environment to a data source" (Williamson, Abstract, lines 1-2). Peddada, cited by the Examiner for its drag-and-drop interface, is directed to "providing a graphical user interface on a client network device" (Peddada, Abstract, lines 1-2). Gupta, cited by the Examiner its highlighting of a set or sets of annotations grouped together for execution, is directed to "[a]n annotation server [that] uses a hierarchical annotation storage structure to maintain a correspondence between a plurality of multimedia stream annotations and a hierarchically higher group identifier" (Gupta, Abstract, lines 1-2). O'Donnell, cited by the Examiner for disclosing a means for receiving a list of system identifiers of a particular computer system, is directed to a "method for performing parallel management operations including and deleting computer systems" (O'Donnell, Title; see also Abstract, lines 1-12). Chow, cited by the Examiner for disclosing the binding of statements to a database application, is directed to "and integrated compiler for compiling SQL3 control statements having procedural, i.e., control, information packaged together with query, i.e., non-procedural, statements" (Chow, Abstract, lines 1-4). However, none of these references disclose inventions directed or related to a visual programming environment, and thus, not surprisingly, all of these references and Farrell, alone

or in combination, completely fail to teach or suggest a graphical object representing the actual code of the source code module in a visual programming environment.

In contrast, the invention of the present Application is directed to "an integrated development environment wherein there is a tight coupling between the design surface providing a visual representation of the various physical and logical entities in the software model and the underlying code structures that support the entities" such that "[e]very object defined within the design surface is capable of being mapped **directly** to an underlying code structure" (Application Specification, page 2, line 27 to page 3, line 1) (emphasis added). The net result is that, in the invention of the present Application, "[t]he model is a graphical representation of **the actual code**" (Application Specification, page 2, line 5) (emphasis added). The clarify this point in the claims, the aforementioned amendments have been made.

In order to establish a prima facie case of obviousness, three basic criteria must be met. First there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second there must be a reasonable expectation of success. Finally the prior art reference (or references when combined) must teach or suggest all the claim elements. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and cannot be based on applicant's disclosure. (MPEP §§ 2142, 2143.)

In regard to the third criteria, Applicant(s) respectfully submit that Farrell in view Washburn, Bier, Williamson, Peddada, Gupta, O'Donnell, Chow, or any combination thereof, fails to teach or suggest all the claim elements present in Claims 1, 15, and 22, upon which Claims 3-7 and 8-14; 16-21; and 23-28 respectively depend. Specifically, in the context of a visual programming environment, nowhere do Farrell, Washburn, Bier, Williamson, Peddada, Gupta, O'Donnell, and Chow, alone or in combination, teach or suggest the element of a graphical object representing actual code of the source code module such that every object defined within the design surface is capable of being mapped directly to the underlying source code structure. Consequently, these references fails to teach or suggest all the claim elements necessary to anticipate the invention of Claims 1, 15, and 22 under 35 U.S.C. § 103(a), and Applicant(s) therefore respectfully request that the rejection of Claims 15 and 22 under this section be withdrawn. Moreover, given that claims which depend upon an allowable claim are

also allowable, and since Claims 3-7 and 9-14; 16-21; and 23-28 depend upon allowable Claims 1, 15, and 22 respectively, Applicant(s) respectfully request that the rejections as to Claims 3-7 and 9-14 under this section be withdrawn and that these claims also be allowed to issue.

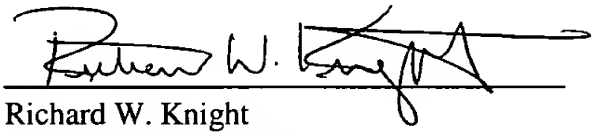
[Remainder of Page Intentionally Left Blank]

CONCLUSION

Based on the reasons and rationale set forth herein, Applicants respectfully submit that the objections and rejections have been overcome and, accordingly, Applicants request that the objections and rejections be withdrawn and that the claims be allowed to issue. Should the Examiner have any questions, comments, or suggestions that would expedite the prosecution of the present case to allowance, Applicants' undersigned representative earnestly requests a telephone conference at (206) 332-1394.

Respectfully submitted,

Date: January 15, 2004



Richard W. Knight
Registration No. 42,751

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439